Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-7. (Cancelled).

Claim 8. (Previously presented) A computer system, comprising:

a memory;

a superscalar microprocessor for processing instructions; and

a bus coupled between the memory and the microprocessor;

wherein the microprocessor includes:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

an instruction buffer coupled to receive and buffer fetched instructions from the instruction fetch unit;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a resource identifying circuit configured to concurrently identify execution resources for more than one of a plurality of buffered instructions, the identified execution resources for each of the buffered instructions including a functional unit capable of executing the instruction and a register file entry corresponding to a

source of an operand for the instruction, thereby making a plurality of instructions concurrently available for execution;

an issue control circuit coupled to the resource identifying circuit and configured to concurrently issue more than one of a plurality of available instructions to the functional units for execution, based on availability of the execution resources identified by the resource identifying circuit and without regard to the sequential program order;

a plurality of data routing paths coupled between the plurality of functional units and the register file and configured to concurrently transfer result data from more than one of the plurality of functional units to the register file; and

bypass control logic coupled to the plurality of data routing paths and configured to distribute result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via an alternate data path that bypasses the register file, wherein distributing result data via the alternate data path occurs concurrently with transferring result data to the register file.

Claim 9. (Previously presented) The system of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that an integer result from the integer functional unit is distributed to the floating-point functional unit via the alternate data path.

Claim 10. (Previously presented) The system of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that a floating-point result from the floating-point functional unit is distributed to the integer functional unit via the alternate data path.

Claim 11. (Previously presented) The system of claim 8, wherein the microprocessor further includes:

operand data routing paths coupled between the register file and the functional units and configured to concurrently transfer operand data to more than one of the functional units.

Claim 12. (Previously presented) The system of claim 11, wherein the operand data routing paths transfer operand data directly from the register file to the functional units.

Claim 13. (Cancelled).

Claim 14. (Currently amended) A superscalar microprocessor for processing instructions, the microprocessor comprising:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

an instruction buffer coupled to receive and buffer fetched instructions from the instruction fetch unit;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a resource identifying circuit, disposed at a stage subsequent to said instruction buffer, configured to concurrently identify execution resources for a plurality of buffered instructions, thereby making a plurality of instructions concurrently available for issue, wherein the identified execution resources for each of the available instructions ~~including~~ include a functional unit capable of executing the instruction and a register file entry corresponding to a source of an operand for the instruction; and

an issue control circuit coupled to the resource identifying circuit and configured to concurrently issue more than one the available instructions to the functional units for execution, based on availability of the identified execution resources for each instruction and without regard to the sequential program order.

Claim 15. (Previously presented) The microprocessor of claim 14 wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit.

Claim 16. (Previously presented) The microprocessor of claim 14, further comprising:

operand data routing paths coupled between the register file and the functional units and configured to concurrently transfer operand data to more than one of the functional units.

Claim 17. (Cancelled).

Claim 18. (Previously presented) A method for processing instructions in a superscalar microprocessor, the method comprising:

fetching instructions from an instruction store according to a sequential program order;

buffering a plurality of fetched instructions in an instruction buffer;

concurrently identifying execution resources, by a resource identifying circuit disposed at a stage subsequent to said instruction buffer, for more than one of a plurality of buffered instructions, the identified execution resources for each of the more than one of the plurality of buffered instructions including a functional unit capable of executing the instruction and a register file entry corresponding to a source of an operand for the instruction;

concurrently making available for execution a plurality of instructions for which execution resources are identified;

concurrently issuing more than one of the plurality of available instructions for execution by a plurality of functional units, based on availability of the identified execution resources for each available instruction and without regard to the sequential program order;

executing the issued instructions in the plurality of functional units, thereby generating result data; and

transferring the result data from the functional units to a register file, the register file including a plurality of entries, wherein each of the plurality of entries is accessible by reference to a respective location in the register file.

Claim 19. (Previously presented) The method of claim 18 wherein:

the plurality of functional units includes an integer functional unit and a floating point functional unit.

Claim 20. (Previously presented) The method of claim 18, further comprising:

concurrently transferring operand data from the register file to more than one of the functional units via a plurality of operand data routing paths.

Claims 21-30. (Cancelled).

Claim 31. (Previously presented) A computer system, comprising:

a memory;

a superscalar microprocessor for processing instructions; and

a bus coupled between the memory and the microprocessor;

wherein the microprocessor includes:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

an instruction buffer coupled to receive fetched instructions from the instruction fetch unit and configured to buffer a plurality of fetched instructions;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a resource identifying circuit, disposed at a stage subsequent to said instruction buffer, configured to concurrently identify execution resources for a plurality of buffered instructions, the identified execution resources for each of the buffered instructions including a functional unit capable of executing the instruction and a register file entry corresponding to a source of an operand for the instruction, thereby making a plurality of instructions concurrently available for execution; and

an issue control circuit coupled to the resource identifying circuit and configured to concurrently issue more than one of the available instructions to the functional units for execution, based on availability of the execution resources identified by the resource identifying circuit and without regard to the sequential program order.

Claim 32. (Previously presented) The computer system of claim 31 wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit.

Claim 33. (Previously presented) The computer system of claim 31, further comprising:

operand data routing paths coupled between the register file and the functional units and configured to concurrently transfer operand data to more than one of the functional units.

Claims 34-43. (Cancelled)

Claim 44. (Previously presented) The system of claim 8, further comprising retirement control logic coupled to the register file and configured to concurrently retire a plurality of instructions according to the sequential program order.

Claim 45. (Previously presented) The system of claim 44, wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

and wherein the retirement control logic is further configured such that when an instruction is retired, corresponding result data is transferred from the temporary buffer to the retired register array.

Claim 46. (Previously presented) The microprocessor of claim 14, further comprising:

retirement control logic coupled to the register file and configured to retire a plurality of instructions according to the sequential program order.

Claim 47. (Previously presented) The microprocessor of claim 46 wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

wherein the retirement control logic is further configured such that when an instruction is retired, corresponding result data is transferred from the temporary buffer to the retired register array.

Claim 48. (Previously presented) The method of claim 18, further comprising retiring instructions according to the sequential program order.

Claim 49. (Previously presented) The method of claim 48 wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

wherein the retiring an instruction includes transferring corresponding result data from the temporary buffer to the retired register array.

Claim 50. (Previously presented) The computer system of claim 31, further comprising retirement control logic coupled to the register file and configured to retire a plurality of instructions according to the sequential program order.

Claim 51. (Previously presented) The computer system of claim 50 wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

wherein the retirement control logic is further configured such that when an instruction is retired, corresponding result data is transferred from the temporary buffer to the retired register array.

Claim 52. (Previously presented) The system of claim 8, wherein the resource identifying circuit is further configured to concurrently identify execution resources for a first one and a second one of the plurality of buffered instructions, wherein the second one of the instructions has a data dependency on the first one of the instructions.